

# Gnuplot – wprowadzenie v. 2.0

M. Kopciuszyński

22 stycznia 2019

## 1 Wstęp

Gnuplot to zaawansowany program do tworzenia wykresów dwu- i trójwymiarowych. Jest dostępny zarówno na platformę Linux jak i Windows. Wykresy tworzone przy pomocy tego programu spełniają standardy przyjęte przy publikowaniu wyników w czasopiśmie naukowych.

Więcej informacji na temat oprogramowania gnuplot (oraz pliki do pobrania) można znaleźć na stronie domowej projektu: [www.gnuplot.info](http://www.gnuplot.info).

### 1.1 Gnuplot w systemie Windows

Aby zainstalować gnuplota w systemie Windows należy pobrać plik instalacyjny ze strony: [sourceforge.net/projects/gnuplot/files/gnuplot/](http://sourceforge.net/projects/gnuplot/files/gnuplot/).

Po zainstalowaniu programu w menu start znajdziemy nowy skrót. Jego uruchomienie otwiera okno z konsolą. Program zawiera rozbudowane menu ułatwiające wprowadzanie komend.

### 1.2 Gnuplot w systemie Linux

Ponieważ program gnuplot został pierwotnie napisany dla systemu Linux, jego instalacja w tym systemie jest bardzo prosta. Wystarczy przy użyciu menagera pakietów zainstalować pakiet gnuplot, bądź wydać polecenie w terminalu (dla systemu OpenSUSE):

```
zypper in gnuplot
```

bądź w systemie Linux Mint / Ubuntu:

```
sudo apt-get install gnuplot
```

Aby uruchomić zainstalowany program należy w wierszu poleceń wpisać komendę:

```
gnuplot
```

**Uwaga!** Niniejsza instrukcja została przygotowana z myślą o używaniu gnuplota w systemie Linux, jednak polecenia tego programu będą takie same w systemie Windows. Dodatkowo w systemie Windows gnuplot posiada prosty interfejs graficzny, który umożliwia ustawienia niektórych właściwości wykresu bez znajomości poleceń. Pewne różnice mogą pojawić się przy zapisie wykresów do plików graficznych.

## 2 Korzystanie z programu gnuplot – pierwsze kroki

Istnieje wiele nakładek graficznych pozwalających korzystać z gnuplota w sposób uproszczony, jednak zasadniczo program ten obsługiwany jest przy pomocy komend wydawanych w wierszu poleceń.

Po uruchomieniu programu możemy wyświetlić listę dostępnych opcji wpisując komendę:

```
1 help
```

---

Następnie należy postępować zgodnie ze wskazówkami ukazującymi się na ekranie i wybrać interesujący nas temat. Można również uzyskać pomoc dla konkretnej funkcji wpisując np.:

```
1 help plot
```

---

## 2.1 Polecenie *plot*

Podstawowym poleceniem gnuplota jest komenda *plot*, która pozwala na wygenerowanie wykresu funkcji matematycznej bądź wizualizację danych zawartych w pliku tekstowym.

W celu przetestowania tej funkcji możemy wykonać następujące polecenie:

---

```
1 plot sin(x)
```

---

Po jego wykonaniu powinien się ukazać wykres funkcji sinus. Wykres zostanie wyświetlony w standardowym zakresie  $x$  od -10 do 10.

Program gnuplot automatycznie generuje punkty  $x$  w podanym zakresie, a następnie oblicza wartości funkcji, która ma zostać wyświetlona i na podstawie wyznaczonych wartości  $y$  rysuje wykres. Domyślnie tworzone jest 100 próbek (punktów). W niektórych przypadkach może się okazać, że jest to zbyt mała liczba. Wówczas możemy zwiększyć ilość punktów np. do 500 używając polecenia:

---

```
1 set samples 500
```

---

Po zmianie każdego parametru (w tym przypadku był to parametr *samples*) należy zastosować polecenie *replot*. Zazwyczaj jednak najpierw ustalamy wszystkie parametry wykresu, a komenda *plot* wydawana jest na końcu.

## 2.2 Skróty klawiszowe

Po wpisaniu polecenia *plot* (ewentualnie *replot*) pojawia się okno z wykresem. Jeśli jest ono aktywne, to możliwe jest użycie skrótów klawiszowych, które umożliwiają np. włączenie siatki lub ustawienie logarytmicznej skali osi. Listę skrótów można znaleźć na stronie:

<http://www.cs.unc.edu/~jeffay/dirt/FAQ/gnuplot.html>.

Skróty klawiszowe są wykorzystywane stosunkowo rzadko, jednak trzeba mieć na uwadze, że jeśli okno z wykresem jest aktywne to kolejne polecenie może zostać zinterpretowane jako seria skrótów. Będzie to skutkowało niezamierzoną modyfikacją wykresu.

## 2.3 Ustawienia osi

Parametry wykresu można zmienić przy pomocy komend, których przykładowe zastosowanie zamieszczono poniżej.

---

```
1 # komendy zaczynające się znakiem (#) są ignorowane
2 plot sin(x) # wyświetlamy wykres sin(x)
3 set title "Tytuł wykresu" # ustawienie tytułu wykresu
4 set xlabel "Opis osi x" # ustawienie opisu osi x
5 set ylabel "Opis osi y" # ustawienie opisu osi y
6 set grid # wyświetlenie siatki
7 set xrange [0:20] # zakres osi x
8 set yrange [-2:2] # zakres osi y
9 set autoscale # automatyczne skalowanie osi
10 set xtics ("0" 0, "180" pi, "360" 2*pi) # położenie znaczników osi x
11 replot # ponowne narysowanie wykresu - oświecenie
```

---

## 2.4 Definiowanie funkcji

W gnuplocie można definiować własne funkcje. Przykładowo, można zdefiniować funkcję kwadratową w postaci:

$$y(x) = x^2 - x + 1 \tag{1}$$

W tym celu należy zastosować poniższe polecenia:

---

```
1 y(x) = x**2-x+1 # potęgowanie wykonywane przez operator **
2 plot y(x) # wyświetlamy uzyskana funkcje
```

---

Możliwe jest również definiowanie własnych zmiennych i ich późniejsze użycie:

---

```

1 a = 2
2 b = -1
3 c = 1
4 y(x) = a*x**2+b*x+c
5 plot y(x) # wyswietlamy uzyskana funkcje

```

---

W każdym momencie można zmodyfikować wartości zmiennych i narysować wykres ponownie. W tym celu wpisujemy kolejno komendy:

---

```

1 a = 1.4
2 replot # w celu ponownego narysowania wykresu

```

---

Zmienne muszą być zawsze zadeklarowane przed ich wykorzystaniem.

## 2.5 Zapisywanie wykresów

Obsługa programu gnuplot działa na zasadzie wejścia-wyjścia. Wejściem jest terminal, w którym możemy wpisywać polecenia bądź uruchamiać polecenia zapisane w tzw. *pliku wsadowym* (skrypcie). Domyślnym wyjściem jest serwer graficzny x11 – czyli okienko, w którym pojawia się wykres. Aby zapisać wykres do pliku graficznego należy najpierw ustawić typ wyjścia (typ terminala), a następnie wskazać nazwę pliku, w którym wykres zostanie zapisany. Dopiero wtedy wydajemy polecenie *plot*.

W praktyce, aby zapisać wykres do pliku png należy wpisać następujące komendy:

---

```

1 # Uwaga! Kolejnosć wpisywania poleceń jest ważna!
2 #####
3 # 1. Ustawiamy format wyjściowy (typ terminala i rozdzielczosć)
4 #####
5 set term pngcairo size 1024,768
6 #####
7 # 2. Ustawiamy nazwę pliku wyjściowego
8 #####
9 set output "wykres.png"
10 #####
11 # 3. Generujemy wykres
12 #####
13 plot sin(x)

```

---

Jeśli ustawiony jest inny terminal niż domyślny (X11) to na ekranie nie zobaczymy żadnego wykresu, gdyż trafia on bezpośrednio do pliku graficznego. Tego typu rozwiązanie może wydawać się dziwaczne, jednak jego moc poznajemy gdy zachodzi potrzeba przygotowania dużej ilości wykresów zawierających podobne dane.

Aby wrócić do „normalnego” trybu pracy możemy wpisać polecenie<sup>1</sup> *set term x11*.

## 3 Używanie skryptów w programie gnuplot

Skrypt to plik tekstowy zawierający w kolejnych wierszach komendy programu. Plik taki może zostać odczytany przez program gnuplot, dzięki czemu nie musimy wpisywać kolejnych poleceń w terminalu. Pozwala to na zautomatyzowanie powtarzalnych czynności takich jak ustawienie parametrów wykresu, opisu i zakresu osi.

### 3.1 Konsola w systemie Linux

Gnuplot jest uruchamiany w systemie Linux z poziomu terminala, dlatego do jego obsługi przydatna jest znajomość podstawowych komend konsoli (terminala) Linux.

Po uruchomieniu konsoli systemu Linux wiersz poleceń rozpoczyna się zwykle od następującej informacji:

```

user:~$
# lub
user:/home/student/dokumenty $

```

---

<sup>1</sup>To polecenie zależy od wersji zainstalowanego gnuplota. W systemie Linux często używa się polecenia *set term qt*.

po wykonaniu polecenia *gnuplot* (uruchomieniu tego programu) na początku każdej nowej linii pojawia się poniższy wpis:

```
1 gnuplot>
```

Aby wrócić do konsoli systemowej należy wpisać polecenie *quit*. Jest to istotne z tego powodu, że polecenia programu *gnuplot* nie będą działać w konsoli systemowej (i na odwrót)<sup>2</sup>.

Dla lepszej przejrzystości przyjęto konwencję, że polecenia konsoli systemowej są w niniejszej instrukcji wyrównane do lewej, natomiast polecenia programu *gnuplot* są zapisane z dodatkowym wcięciem, zawierają numerację wierszy oraz są oddzielone od tekstu przy pomocy poziomych linii – co widać na powyższych przykładach.

Do zmiany katalogu oraz wyświetlenia jego zawartości wykorzystamy poniższe komendy:

```
cd folder
ls
```

Pierwsza z nich: *cd folder* pozwala na zmianę katalogu na *folder*, druga komenda: *ls* wyświetla w terminalu zawartość katalogu, w którym się znajdujemy. Dodatkowo przydatne będą komendy:

```
touch nawaPliku
mkdir nazwaKatalogu
pwd
```

Polecenie *touch* pozwala utworzyć plik tekstowy, *mkdir* tworzy nowy katalog o podanej nazwie, *pwd* wyświetla naszą obecną lokalizację.

## 3.2 Uruchamianie skryptów w gnuplocie

Skrypty gnuplota zapisujemy w plikach tekstowych, które mogą mieć dowolne rozszerzenie, jednak w systemie Windows używamy rozszerzeń \*.txt lub \*.pg. Najlepiej jest na potrzeby każdego wykresu (lub grupy wykresów) utworzyć osobny katalog (roboczy), w którym znajdą się wszystkie potrzebne pliki wraz ze skryptem.

Przykładowy skrypt może mieć taką treść:

```
1 reset
2 set term pngcairo
3 set output "wykres.png"
4 plot sin(x)
```

Istnieją dwa sposoby uruchomienia skryptów w programie gnuplot. W obu przypadkach musimy znajdować się w naszym katalogu roboczym zawierającym plik skryptu (obecną lokalizację można sprawdzić przy pomocy polecenia *pwd*).

Pierwsza metoda polega na uruchomieniu gnuplota, a następnie załadowaniu do uruchomionego programu skryptu (*plot1.pg*) przy pomocy polecenia:

```
1 load "plot1.pg"
```

Po załadowaniu skryptu w katalogu roboczym powinien pojawić się plik o nazwie *wykres.png* zawierający wykres funkcji  $\sin(x)$ .

Skrypt można również załadować do programu *gnuplot* od razu podczas jego uruchamiania. Należy wówczas w terminalu systemu Linux wpisać polecenie:

```
gnuplot plot1.pg
```

Druga metoda ma tę przewagę, że gnuplot jest uruchamiany tylko na czas generowania wykresów.

<sup>2</sup>Chociaż są pewne wyjątki, np.: polecenie *pwd* działa zarówno w programie gnuplot jak i konsoli systemu Linux.

## 4 Wymiary wykresu i rozmiar czcionek

Poniżej zamieszczono komendy (umieszczone w skrypcie), które można wykorzystać do ustalania parametrów wykresu takich jak: rozmiar czcionki, czy rozmiar wykresu.

---

```
1 reset
2 set xlabel "Opis osi x" # ustawienie opisu osi x
3 set ylabel "Opis osi y" # ustawienie opisu osi y
4 set xrange [0:20] # zakres osi x
5 set yrange [-2:2] # zakres osi y
6
7 # ustawienie terminala / czcionki / rozmiaru wykresu
8 set term pdf font "Verdana,15" size 5,2
9 # możemy ustawić niezależnie czcionki dla osi x i y oraz legendy
10 set key font ",20" # rozmiar czcionki legenda
11 set tics font "Verdana,12" # rozmiar czcionki os x i y
12
13 set output "wykresCW2a.pdf"
14
15 plot sin(x)
```

---

Rozmiar czcionki może zostać ustalony dla całego wykresu przy pomocy polecenia zawartego w linii 8. Można również ustalić niezależnie rozmiar czcionek etykiet osi oraz legendy, co jest wykonywane przy pomocy poleceń w liniach 10 i 11. Proszę zwrócić uwagę, że można zdefiniować zarówno typ czcionki i jej rozmiar, jak również sam rozmiar czcionki. W drugim przypadku używamy składni z przecinkiem (patrz linia 10): *font " ,10"*.

Rozmiar wykresu definiowany jest poleceniem *size* w linii 8. Należy jednak pamiętać, że ma on różne znaczenie w zależności od wyboru typu terminala. Dla wykresów wektorowych (pdf, eps) będzie to zwykle rozmiar wyrażony w calach, dla wykresów w formatach rastrowych będzie to liczba pikseli.

## 5 Wykresy danych z pliku tekstowego

Zasadniczym zdaniem gnuplota jest rysowanie wykresów w oparciu o pliki tekstowe zawierające dane. Plik tekstowy powinien mieć określoną strukturę. W kolejnych liniach będą znajdować się punkty pomiarowe. Najpierw podana jest współrzędna *x*, następnie współrzędna *y*. Możliwe jest również przetwarzanie plików zawierających więcej kolumn. Kolejne kolumny mogą zawierać informację na temat niepewności pomiarowych.

Przykładowy plik z danymi może wyglądać następująco:

```
20.2682 24136.005
20.3182 24279.511
20.3682 24212.279
20.4182 23742.821
```

Używając poniższego skryptu wygenerować plik *wykresDane.pdf* zawierający wykres danych zawartych w pliku o nazwie *dane.dat* komendy:

---

```
1 set term pdf
2 set output "wykresDane.pdf"
3 plot "dane.dat"
```

---

### 5.1 Struktura danych wejściowych

Plik wejściowy może zawierać przykładowo trzy kolumny. Wówczas pierwsza z nich będzie traktowana jako wartości *x*, druga – wartości *y*, natomiast trzecia kolumna może zawierać wartości błędów dla *y*.

Gnuplot domyślnie traktuje pierwsze dwie kolumny jako wartości *x* oraz *y*. Możliwe jest jednak wymuszenie użycia danych zawartych w innych kolumnach. Można tego dokonać używając polecenia:

---

```
1 plot 'daneCW3.dat' using 1:3
```

---

W takim przypadku gnuplot ignoruje dane z 2 kolumny, a używa jedynie 1 i 3.

Jeśli chcemy aby trzecia kolumna została wykorzystana do wykreślenia znaczników błędu, możemy użyć zmodyfikowanej komendy plot:

---

```
1 plot 'daneCW3.dat' with yerrorbars
```

---

Analogicznie możemy narysować znaczniki błędu dla wartości  $x$ , a także dla obu wartości:

---

```
1 plot 'daneCW3.dat' with xerrorbars
2 # lub
3 plot 'daneCW3.dat' with yerrorbars
4 # lub
5 plot 'daneCW3.dat' with errorbars
```

---

W trzecim przypadku plik musi zawierać już cztery kolumny danych.

## 5.2 Zmiana typu znaczników punktów

Gnuplot domyślnie ustawia znaczniki w postaci fioletowych plusów  $+$ . Mamy jednak do dyspozycji całą gamę kolorów jak również typów znaczników. Poniższe polecenia rysują ten sam wykres przy użyciu różnych znaczników:

---

```
1 plot 'daneCW3.dat' pt 1 ps 1
2 # lub
3 plot 'daneCW3.dat' pt 2 ps 2
4 # lub
5 plot 'daneCW3.dat' pt 3 ps 3 lc rgb "blue"
```

---

Typ punktu wybieramy przy pomocy komendy *pt* podając numer. Komenda *ps* ustala rozmiar punktów. Kolor punktów można zmienić poleceniem *lc rgb*. Jako argument tego polecenia podajemy nazwę koloru, lub jego kod w notacji szesnastkowej.

Jeśli chcemy szybko przekonać się jakie typy punktów mamy do dyspozycji należy zamiast polecenia *plot* zastosować komendę:

---

```
1 test
```

---

## 5.3 Dopasowanie prostej do danych doświadczalnych

Gnuplot umożliwia dopasowanie dowolnej krzywej (opisanej funkcją matematyczną) do danych doświadczalnych. W celu dopasowania funkcji liniowej (prostej) do danych doświadczalnych należy zastosować następujące polecenia:

---

```
1 f(x)=a*x+b
2 fit f(x) 'daneCW3.dat' via a,b
```

---

Pierwsze polecenie tworzy funkcję liniową  $f(x) = ax + b$ . Kolejne, dopasowuje funkcję  $f(x)$  do danych zawartych w pliku *daneCW3.dat* wykorzystując do tego celu parametry  $a$  oraz  $b$ . Innymi słowy, powyższe polecenia można wykorzystać do wyznaczenia współczynników  $a$  oraz  $b$  funkcji liniowej podobnie jak przy zastosowaniu metody najmniejszych kwadratów.

Gnuplot umożliwia zdefiniowanie niemal dowolnej funkcji matematycznej oraz wykona próbę jej dopasowania przy pomocy zadanych parametrów. Jednak dla bardziej złożonych funkcji dopasowanie może być mało trafne, lub wręcz niemożliwe.

Po wywołaniu komendy *fit* gnuplot utworzy dodatkowy plik tekstowy *fit.log*. Plik ten zawiera raport z dopasowania funkcji do danych doświadczalnych. Można w nim znaleźć informację o wartości dopasowanych parametrów oraz niepewności dopasowania:

Final set of parameters	Asymptotic Standard Error
=====	=====
a = 45.3457	+/- 2.598 (5.729%)
b = -19.0833	+/- 1.315 (6.89%)

## 5.4 Rysowanie dopasowanej funkcji

Po wykonaniu polecenia *fit* funkcja  $f(x)$  posiada już zaktualizowane wartości zmiennych  $a$  oraz  $b$ . Żeby ją zilustrować należy wydać polecenie:

---

```
1 plot f(x)
```

---

Aby oprócz dopasowanej funkcji  $f(x)$  wykreślić również dane z pliku tekstowego, to przy poleceniu `plot` należy podać po przecinku parametry kolejnego wykresu:

---

```
1 plot f(x), 'daneCW3.dat'
```

---

Obie serie danych można dowolnie modyfikować, korzystając z omówionych wcześniej parametrów `ps`, `pt` oraz `lc` np.:

---

```
1 plot f(x) lc rgb 'red' lw 2, 'daneCW3.dat' pt 7 ps 1.5 lc rgb 'blue'
```

---

Dodatkowy parametr `lw` pozwala na ustawienie grubości linii.

Dobrym nawykiem jest zapisywanie parametrów kolejnych wykresów w kolejnych liniach, np.:

---

```
1 plot f(x) lc rgb 'red' lw 2, \  
2 'daneCW3.dat' pt 7 ps 1.5 lc rgb 'blue'
```

---

Wówczas po przecinku trzeba wstawić znak: „\`\"`”.

Pamiętamy jednak, że polecenie `plot` może pojawić się tylko raz w skrypcie<sup>3</sup>.

## 5.5 Wstawianie etykiet

W gnuplocie istnieje możliwość wstawiania własnego tekstu w polu wykresu. W tym celu można użyć polecenia `set label`. Poniżej umieszczony przykładowe zastosowanie tej funkcji.

---

```
1 # etykieta wstawiona w punkcie x = 1, y = 0.5  
2 set label "Napis 1" at 1,0.5  
3  
4 # etykieta w postaci wartosci zmiennej wstawiona w punkcie x = -5, y = 0.5  
5 a = 1.5  
6 set label sprintf("a = %2.2f",a) at -5,0.5  
7 replot
```

---

Polecenie zawarte w linii 6 pozwala na utworzenie etykiety przedstawiającej wartość zmiennej  $a$ , która została przypisana w linii 5. W tym przypadku składnia wygląda trochę nietypowo, gdyż polecenie `sprintf` jest zapożyczone z języka C. Tego polecenia można użyć żeby wstawić na wykresie wartości parametrów  $a$  oraz  $b$  uzyskane wcześniej przy pomocy polecenia `fit`. Dzięki temu nie jest konieczne „ręczne” przepisywanie tej wartości.

## 6 Zakończenie

W niniejszej instrukcji zamieszczono jedynie podstawowe informacje dotyczące programu gnuplot. Powinny być one pomocne do rozpoczęcia pracy z tym programem i jednocześnie wystarczające do przygotowania wykresów na I czy II pracowni fizycznej. Najlepszym sposobem poznania pozostałych możliwości tego programu jest przegląd demonstracyjnych wykresów zamieszczonych na stronie: [http://gnuplot.sourceforge.net/demo\\_5.0/](http://gnuplot.sourceforge.net/demo_5.0/).

Prawdziwe możliwości gnuplota docenimy jednak dopiero wtedy kiedy będzie on używany wspólnie z systemem składu tekstu  $\text{\LaTeX}$ . Najbardziej optymalnym środowiskiem pracy okaże się wówczas system Linux, który umożliwi dodatkowo interakcję gnuplota,  $\text{\LaTeX}$ 'a, jak również innych programów takich jak Python, Octave, Fityk, dzięki wykorzystania skryptów powłoki systemowej (bash).

---

<sup>3</sup>Ta uwaga nie dotyczy sytuacji, gdy w jednym pliku graficznym chcemy zapisać kilka wykresów.